

Desire to be proud of what we do

SOFTWARE CRAFTSMANSHIP



STOP
CODE ROT



LANGLEY AND
TEAM'S MINDSET

HAVE TO DO IT

WRITE BROTHERS
MINDSET

WANT TO DO IT

IN 1982, A THEORY WAS
INTRODUCED BY
SOCIAL SCIENTISTS
JAMES Q. WILSON

WINDOW



LITTER



GRAFFITI



STRUCTURE



SENSE OF ABANDONMENT
BECOMES REALITY



NEGLECTION *ACCELERATES* THE ROT FASTER
THAN ANY OTHER FACTOR.

HOW IS THIS RELATED WITH THE SOFTWARE?

Are the next few questions
sounds familiar?

CHANGES ARE **HARD** TO DO?

SIMPLE CHANGE **IMPACTS**
NUMEROUS MODULES?

IMPLEMENTING SIMPLE
CHANGE TAKES **FOREVER**?

CHANGE IN ONE PLACE
HARMS SOMEWHERE
COMPLETELY UNEXPECTED
AREA?

FIXING A BUG **CAUSES** <N>
MORE?

MODULES ARE
NOT REUSABLE
BECAUSE OF THEIR
DEPENDENCIES?

IT'S EASIER
TO DO **HACKS**
THAN BY THE BOOK?

BINGO!



```
public static String testableH
PageData pageData,
boolean includeSuiteSetup
) throws Exception {
WikiPage wikiPage = pageDat
StringBuffer buffer = new S
if (pageData.hasAttribute("
if (includeSuiteSetup) {
WikiPage suiteSetup =
PageCrawlerImpl.getIn
SuiteResponde
);
if (suiteSetup != null)
WikiPagePath pagePath
suiteSetup.getPageC
String pagePathName =
```



```
public List<in
List<int[]>
for (int[] x
if (x[0] =
list1.ac
return list1
}
```



```
int a = 1;
if (0 == 1
a = 01;
else
1 = 01;
```

CODE ROT HAS BEGUN?



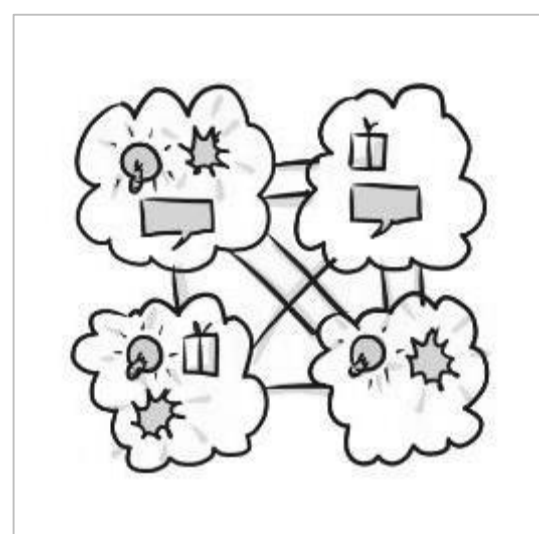
```
/** The child Containers belongi
protected HashMap children = new
/** The processor delay for this
protected int backgroundProcesso
/** The lifecycle event support
protected LifecycleSupport lifec
/** The container event listener
protected ArrayList listeners =
/** The Loader implementation wi
protected Loader loader = null;
```



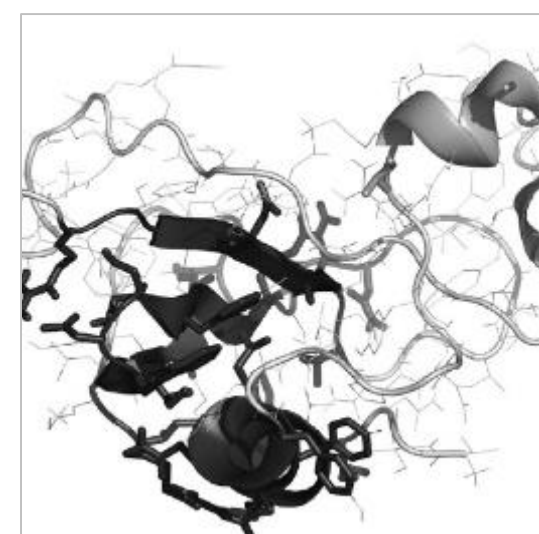
SOFTWARE
EROSION



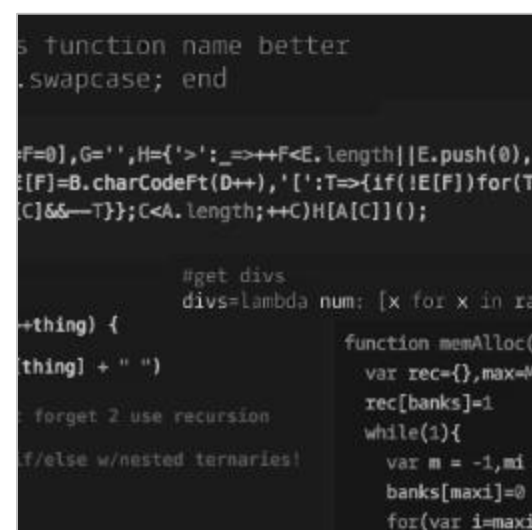
SOFTWARE
DECAY



SOFTWARE
ENTROPY



SOFTWARE
ROT



TECHNICAL
DEBT

WE KNOW
IT BY MANY
NAMES

HOW CAN WE **STOP** IT?



KEEP THE
CODE CLEAN
AT ALL TIMES

WHAT'S THE **CLEAN CODE**?

...DOES **ONE THING** WELL

-- Bjarne Stroustrup

Inventor of C++

...READS LIKE WELL-WRITTEN **PROSE**

-- Grady Booch

Inventor of UML

...IT WAS WRITTEN BY SOMEONE
WHO CARED

-- Michael Feathers

Author of Working effectively with Legacy Code

...FEW PROGRAMMERS KNOW
HOW TO WRITE CODE THAT
HUMAN UNDERSTAND

-- Martin Fowler
Author of Refactoring

...IS WHEN EACH METHOD YOU READ
TURNS OUT TO BE PRETTY MUCH
WHAT **YOU EXPECTED**

-- Ward Cunningham

Inventor of Wiki

Coinventor of Extreme Programming



OUR GOAL IS
NOT TO CODE
BUT TO
COMMUNICATE

1

WHILE MAKING
CHANGES
TO EXISTING CODE

2

WHILE ADDING
NEW CODE

1

WHILE MAKING
CHANGES
TO EXISTING CODE

2

WHILE ADDING
NEW CODE

ANY IDEA WHEN
EXISTING CODE
BECOMES **LEGACY**?

THE MOMENT WE
STOP REFACTORING
OR **STOP CARING**

MAINTENANCE **START** THE MOMENT
WE **WRITE** THE CODE

[illegible][illegible][illegible]

“DON'T PANIC
I KNOW LITTLE REFACTORING
LET'S DO IT!!”

102
COLUMNS

100+
DEPENDENCIES

```
CREATE PROCEDURE dbo.prv_BundleFilter_set_old
```

```
@valueDate DATETIME ,  
@asOfDate DATETIME ,  
@idBundle INT = NULL ,  
@localizationSG INT = NULL ,  
@idFOCode INT = NULL ,  
@isMigrated BIT = NULL ,  
@idNCRStatus INT = NULL ,  
@idCollateralStructure INT = NULL ,  
@isActive INT = 1 ,  
@idProfile INT = NULL
```

AS

```
DECLARE @foCodesTable TABLE
```

```
(  
    idFrontOfficeCode INT  
)
```

```
INSERT INTO @foCodesTable
```

```
SELECT *  
FROM fn_GetDistinctFOCodes(@idProfile)
```

```
DECLARE @CombinedFacility TABLE
```

```
(  
    idBundle INT NULL INDEX IX_idbundle,  
    idBBMultiFacilityHeader INT NULL ,  
    isActive BIT  
)
```

```
CREATE FUNCTION dbo.ts_Bundle_create  
(  
)  
-- the @Summary variable table is central to the function and being filled step by step  
RETURNS @Summary table (  
    idBundle int null INDEX ide_idBundle  
    ,idBundleUp int null  
    ,lft int null  
    ,rgt int null  
    ,idTree int null  
    ,idOutstandingValo int null  
    ,idOutstandingValoEOD int null  
    ,idValo int null  
    ,valueDate datetime null  
    ,asOfDate datetime null  
    ,idItem int null  
    ,type varchar(50) null  
    ,idKey varchar(50) null  
    ,label varchar(255) null  
    ,callcode char(3) null  
    ,callcodeEOD char(3) null  
    ,idLocalID int null  
    ,localID varchar(255) null  
    ,idFrontOfficeCode int null  
    ,frontOfficeCode varchar(255) null  
    ,isMigrated bit null  
    ,idNCRStatus int null  
  
    -- Global Summary Information  
    ,limitAmount float null  
    ,limitAmountEOD float null  
    ,collateralAmount float null  
    ,grossCollateralAmount float null  
    ,outstandingEODAmount float null  
    -- UNDONE by AF : valo outstanding(to be removed) ,outstandingIntradayAmount float null  
    ,outstandingCurrentAmount float null  
    ,outstandingTheoreticalAmount float null -- relativ aux BB  
    ,facilityAvailableEODAmount float null -- (limit - outstanding)  
    ,multiFacilityAvailableEODAmount float null -- (limit - outstanding)  
    ,collatPositionEODAmount float null -- (collat - outstanding)  
    ,availableForDrawingEODAmount float null -- min(facilityAvailableAmount, availableFor  
    ,facilityAvailableCurrentAmount float null -- (limit - outstanding)  
    ,multiFacilityAvailableCurrentAmount float null -- (limit - outstanding)  
    ,collatPositionCurrentAmount float null -- (collat - outstanding)  
    ,availableForDrawingCurrentAmount float null -- min(facilityAvailableAmount, availa)  
  
    -- SG Summary Information  
    ,SGLimitAmount float null  
    ,SGLimitAmountEOD float null  
    ,SGCollateralAmount float null  
    ,SGGrossCollateralAmount float null  
    ,SGCurrentCollateralAmount float null -- just for shared BB current collat  
    ,SGCurrentGrossCollateralAmount float null -- just for shared BB current collat  
    ,SGoutstandingEODAmount float null  
    -- UNDONE by AF : valo outstanding(to be removed) ,SGoutstandingIntradayAmount float  
    ,SGoutstandingCurrentAmount float null  
    ,SGoutstandingTheoreticalAmount float null  
    ,SGFacilityAvailableEODAmount float null -- (limit - outstanding)  
    ,SGMultiFacilityAvailableEODAmount float null -- (limit - outstanding)  
    ,SGCollatPositionEODAmount float null -- (collat - outstanding)  
    ,SGAvailableForDrawingEODAmount float null -- min(facilityAvailableAmount, available  
    ,SGFacilityAvailableCurrentAmount float null -- (limit - outstanding)  
    ,SGMultiFacilityAvailableCurrentAmount float null -- (limit - outstanding)  
    ,SGCollatPositionCurrentAmount float null -- (collat - outstanding)  
    ,SGAvailableForDrawingCurrentAmount float null -- min(facilityAvailableAmount, availa)  
  
    ,minNCR float null  
    ,imperfectlySecuredLimit float null  
  
    ,perfectlySecuredAmount float null  
    ,imperfectlySecuredAvailable float null  
    ,unsecuredAmount float null  
  
    ,perfectlySecuredAmountEOD float null  
    ,imperfectlySecuredAvailableEOD float null  
    ,unsecuredAmountEOD float null  
  
    ,SGperfectlySecuredAmount float null  
    ,SGimperfectlySecuredAvailable float null  
    ,SGunsecuredAmount float null  
  
    ,SGperfectlySecuredAmountEOD float null  
    ,SGimperfectlySecuredAvailableEOD float null  
    ,SGunsecuredAmountEOD float null  
  
    ,MarginCallAmount float null  
  
    ,hasMarginCallThreshold int null  
    ,thresholdType char(1) null  
    ,thresholdValue float null  
    ,idMarginCallAlertScope int null  
    ,idNCRAlertScope int null  
  
    ,grossEODRatio float null  
    ,netEODRatio float null  
    ,grossCurrentRatio float null  
    ,netCurrentRatio float null  
  
    ,idCollateralStructure int null,  
    isOutstandingSGAmount int null,  
    isLimitSGAmount int null,  
    sharingSG float null,  
    sharingSGEOD float null,  
    sharingSGCurrent float null,  
    sharedBB int null,  
  
    idBBSnapshot int null,  
    setSGRole int null,  
    Cooke float null,  
    SyndicatedCooke float null,  
    OutstandingCooke float null,  
    UnusedPortionCooke float null,  
    committed int null,  
    snapshotdate datetime null,  
  
    expiryDate datetime,  
    maturityDate datetime,  
    effectiveDate DATETIME  
  
    --Added by Anant for itrack 1284178  
    ,initialOutstandingSGAmount float null  
    ,initialOutstandingGlobalAmount float null  
    ,grossIncUSGGlobalCollateralRatio float null  
    ,grossIncUSGGlobalCollateralRatio float null  
    ,grossIncUSGGlobalCollateralAmount float null  
    ,grossIncUSGGlobalCollateralAmount float null  
  
    )  
as  
begin  
return  
end
```

Dependencies	
prv_BundleFilter_set	
io_ExportGlobalCollateralFacilities	
nr_ValuationOutstanding_get	
nr_ValuationWarning_get	
nr_ValuationWarningGreenLight_get	
prv_MultiFacility_get	
rs_AnalyticalDataBalance_get	
rs_BundleLabel_get	
rs_ClientExport	
rs_CollateralDetail_get	
rs_CollateralSummary_get	
rs_CombinedFacilityDashboard_get	
rs_Cooke_get	
rs_CounterpartExposure_CustomerImpact_get	
rs_CustomerDashboard_get	
rs_DocumentDetail_get	
rs_DocumentGreenLight_get	
rs_Facility_getAll	
rs_FacilityParameters	
rs_GetCurrencyRate	
rs_GetSubLimits	
rs_Global_getAllByExpiryDate	
rs_GlobalDashboard_get	
rs_GlobalMTMLossAndGain	
rs_GlobalSummary_get	
rs_LiquidityReport_get	
rs_Monitoring_get	
rs_MultiExposure_get	
rs_NCR_DashBoard	
rs_OutstandingSummary_get	
rs_OutstandingSummaryEXCEL_get	
rs_PortfolioReviewReport_Get	
rs_SalesContract_get	
rs_SpotApproval_get	
rs_Tickler_getActiveByIdBundle	
rs_TransactionDashboard_get	
rs_ValoDetail_getRate	
rs_WarehouseDetail_get	
rs_WarehouseDetailsAndExposure_get	
rs_WarningGreenLight_get	
sup_Data_extraction_at_client_level	
sup_Data_Extraction_At_Facilitylevel	
sup_Data_Extraction_DafPrimaryDetails	
sup_Data_Extraction_DafPurchaseDetails	
sup_Data_Extraction_DafSaleDetails	
ts_AnalyticalData_RemoveEntriesForTransaction	

I WAS STILL
SCRATCHING
THE SURFACE



FOLLOWING THE EXISTING PATTERN	UNFAMILIAR CODE	INEXPERIENCED DEVELOPERS
TIME PRESSURE	NEW TO TEAM	MAKING SMALLEST POSSIBLE CHANGE
LACK OF REGULATIONS	DON'T KNOW WHEN IT HAPPENED	HARD TO UNDERSTAND

WHY
THIS STORED
PROCEDURE
BECAME LIKE THIS?

ATTEMPT 1

SMALL CHANGE & IT FAILED

CODE IS RIGID

ATTEMPT 20+

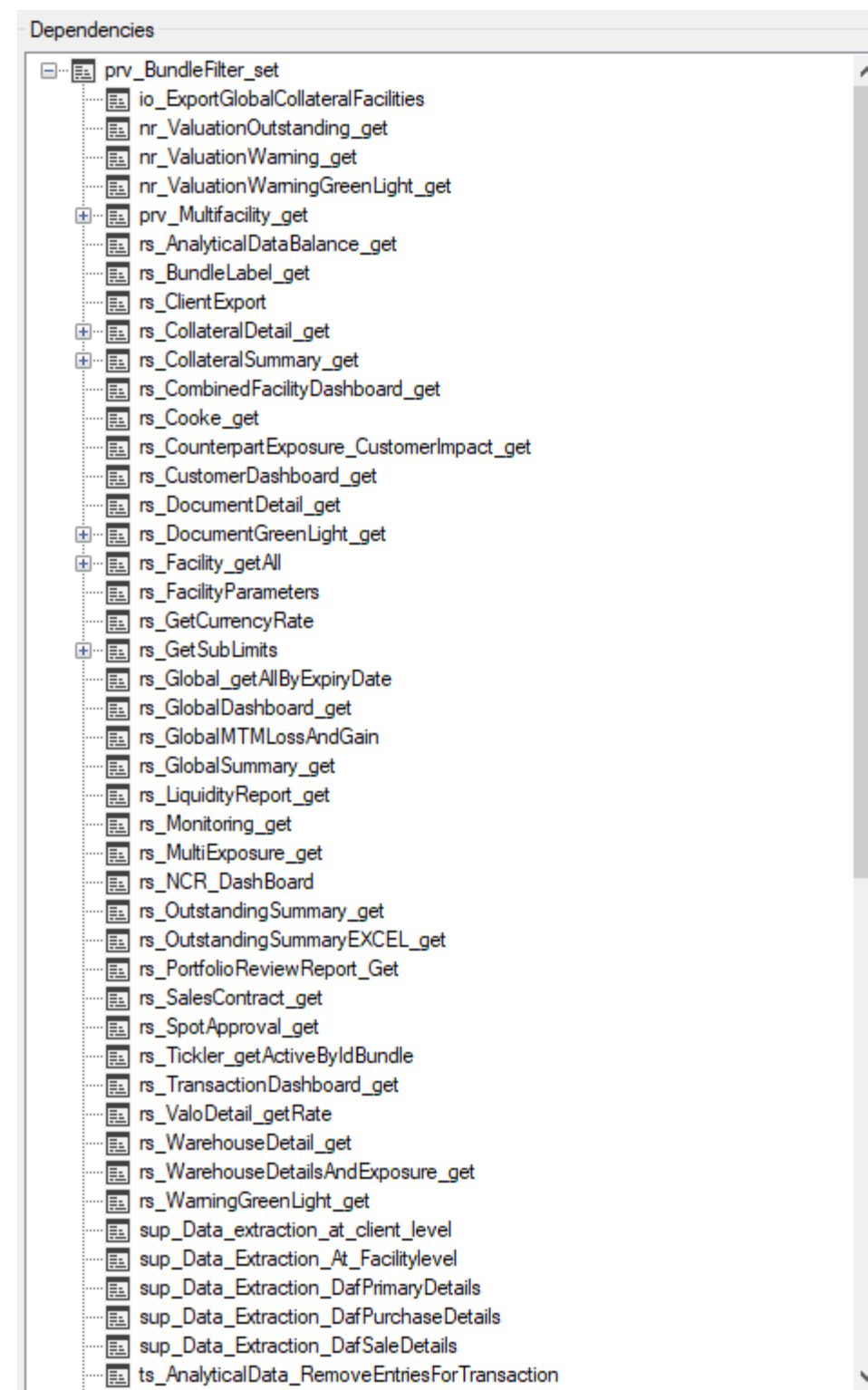
Made changes, it worked!!!
Checked in.

WE RELEASED 😊

After **2 Days** in production
It broke the reports
from completely different part of
application

CODE IS FRAGILE

100+ DEPENDENCIES



When looked deeper into
the code, dependencies
have grown because this
module
**couldn't be reused
partly**

CODE IS IMMOBILE

Module was full of
technical names than
business/domain names

CODE IS NOT READABLE

PERFECT INGREDIENTS FOR?



FEAR





TEST AUTOMATION

UNIT
TESTING

FUNCTIONAL
TESTING

ZJSONPATCH

DIFFY

DATABASE
NONREG PACK

Finds potential bugs in your system
comparing the output of
new code and your *old code*.

STORED PROCEDURE OUTPUT COMPARER

STAND ALONE

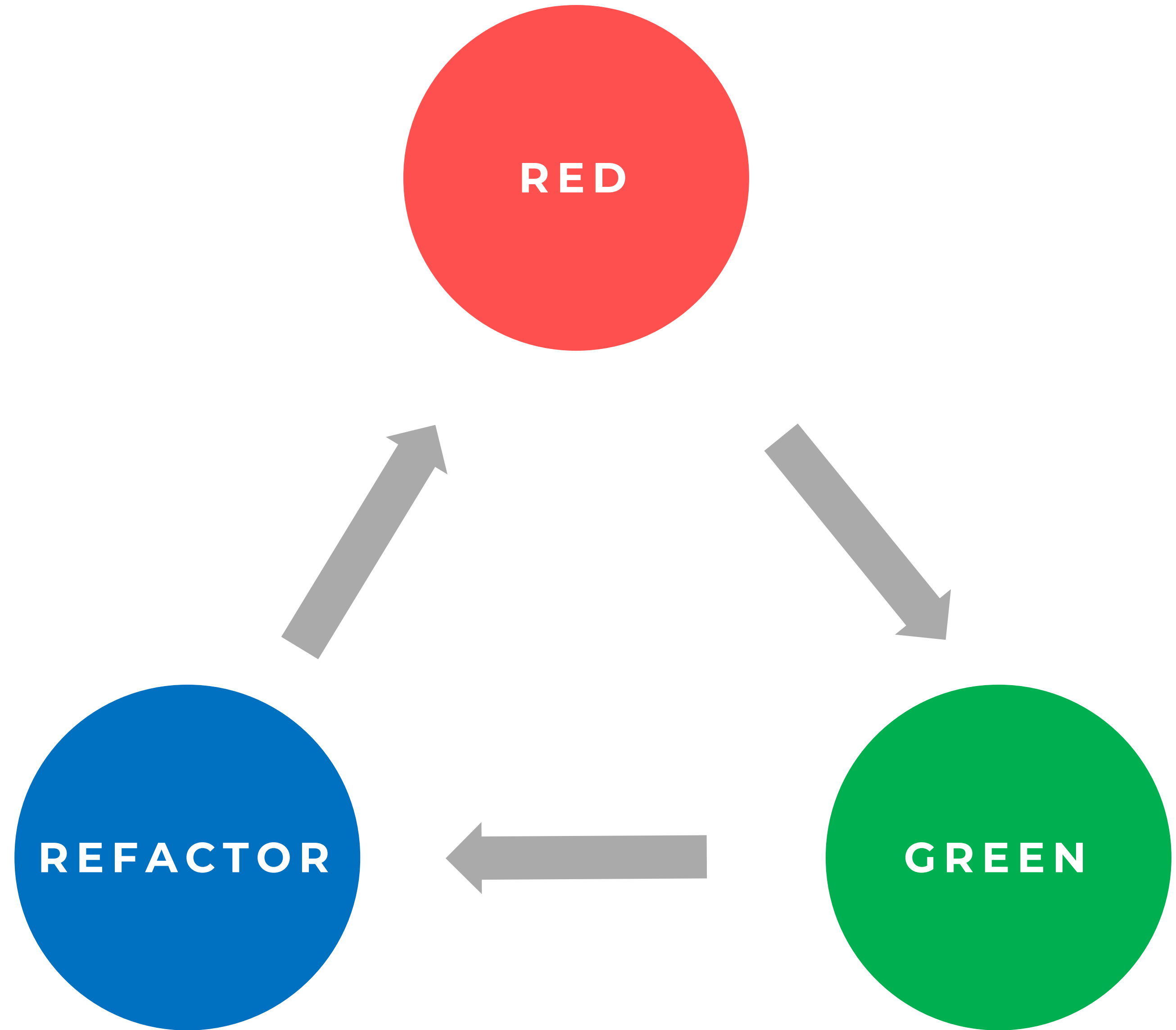
The screenshot shows the SPComparer standalone application. It has a 'Compare From' section with 'Server - Database' set to 'SRVCLDTCT004\MSPARTCTU04 TomcatData_Perf' and 'Stored Procedure' set to 'dbo.ts_FacilityBlockedCash_EconomicMargin'. The 'Compare With' section is identical. The 'Parameters' list includes values for @bundleId, @valueDate, and @bundleId. The 'Results' section shows 'How many times to run?' as 1, 'Parallel Users' as 10, and 'Time for New' as 1616. The 'Avg Diff' is 1112.5. The 'Result' section shows a message: 'Data is not same for Table and for Column idTransactionVersion'. Below this, two tables are displayed side-by-side, showing transaction details. The bottom section shows 'Failed SP with Parameters' with the command: 'dbo.ts_FacilityBlockedCash_EconomicMargin @bundleId = 101, @valueDate = '2019-03-22''.

INTEGRATED WITH CI/CD

The screenshot shows the SPComparer integrated with CI/CD interface. The top navigation bar includes 'Projects', 'Changes', 'Agents', 'Build Queue', and 'Bharat MANE'. The main content area shows a build log for 'FCC/TRA > Tomcat > SP_Comparer > Compile & Compare > #282 (03 Nov 20 01:00)'. The 'Overview' tab is selected, showing a table of results for 'VAL.ref_CommodityQuality_get' and 'VAL.ref_Counterpart_get'. The table has columns: 'Timestamp', 'Result', 'New SP Time', 'Old SP Time', '% Diff', and 'Command'. The results show various errors and warnings, such as 'Data is not same for Table and for Column SG_Rating' and 'Row count is not same for Table'.

Timestamp	Result	New SP Time	Old SP Time	% Diff	Command
11/3/2020 1:02:39 AM	Data is not same for Table and for Column SG_Rating	17	12	-5	VAL.ref_Counterpart_get @idCounterpart = NULL, @filter = 'Z', @activeOnly = NULL
11/3/2020 1:02:39 AM	Data is not same for Table and for Column SG_Rating	15	10	-5	VAL.ref_Counterpart_get @idCounterpart = NULL, @filter = 'Y', @activeOnly = NULL
11/3/2020 1:02:39 AM	Row count is not same for Table	16	13	-3	VAL.ref_Counterpart_get @idCounterpart = NULL, @filter = 'X', @activeOnly = NULL
11/3/2020 1:02:39 AM	Row count is not same for Table	19	14	-5	VAL.ref_Counterpart_get @idCounterpart = NULL, @filter = 'W', @activeOnly = NULL
11/3/2020 1:02:39 AM	Data is not same for Table and for Column SG_Rating	21	24	3	VAL.ref_Counterpart_get @idCounterpart = NULL, @filter = 'V', @activeOnly = NULL
11/3/2020 1:02:39 AM	Row count is not same for Table	20	20	0	VAL.ref_Counterpart_get @idCounterpart = NULL, @filter = 'U', @activeOnly = NULL
11/3/2020 1:02:39 AM	Data is not same for Table and for Column SG_Rating	27	47	20	VAL.ref_Counterpart_get @idCounterpart = NULL, @filter = 'T', @activeOnly = NULL
11/3/2020 1:02:39 AM	Row count is not same for Table	70	118	48	VAL.ref_Counterpart_get @idCounterpart = NULL, @filter = 'S', @activeOnly = NULL
11/3/2020 1:02:39 AM	Row count is not same for Table	45	35	-10	VAL.ref_Counterpart_get @idCounterpart = NULL, @filter = 'R', @activeOnly = NULL
11/3/2020 1:02:39 AM	Row count is not same for Table	16	12	-4	VAL.ref_Counterpart_get @idCounterpart = NULL, @filter = 'Q', @activeOnly = NULL
11/3/2020 1:02:39 AM	Row count is not same for Table	31	58	27	VAL.ref_Counterpart_get @idCounterpart = NULL, @filter = 'P', @activeOnly = NULL

REFACTOR
FEARLESSLY



11.626 SECONDS

01.08 SECONDS

Statement	Est Cost %	Compile Time	Duration	CPU	..
dbo.tmp_testBundleFilter_For_old @valueDate = '2020-11-03 11:59:52', -- date...	100.0%		11,626	468	
EXEC dbo.[prv_BundleFilter_set] @valueDate = @valueDate, -- datetime @a...	99.2%		10,630	468	
EXECUTE sp_executesql @QueryStatementBundleNullBlock, @parameter...	7.5%		1,417	46	
INSERT INTO #bundle1 (idBundle , idBundleUp , lft , rgt , idTree , val...	7.5%	62	68	29	
INSERT INTO @foCodesTableParam SELECT * FROM fn_GetDistinctFOCo...	0.1%	0	185	2	
INSERT INTO #bundle1 (idBundle, idBundleUp, lft, rgt, idTree, valueDat...	16.9%	56	166	66	
DELETE FROM #bundle1 WHERE idBundle IN (SELECT idBundle FROM @...	0.8%	5	134	0	
UPDATE #bundle1 SET type = 'Facility', idItemType = 2, -- Ref_ItemType...	4.4%	33	85	28	
UPDATE B SET type = 'Box', idItemType = 3, -- Ref_ItemType idKey = C...	13.2%	25	77	39	
UPDATE @CombinedFacility SET isActive = 1 WHERE idBBMultiFacilityHea...	0.2%	31	60	28	
UPDATE B SET minNCR = CASE WHEN B.idBundle = F.idBundle AND FC.id...	13.9%	9	39	39	
UPDATE B SET hasMarginCallThreshold = FC.hasMarginCallThreshold, thr...	4.5%	6	23	17	
INSERT #bundle SELECT * FROM #bundle1	7.9%	3	18	9	
UPDATE #bundle1 SET type = 'Customer', idItemType = 1, -- Ref_ItemT...	8.3%	10	12	11	
UPDATE B SET localBL = RLBL.label FROM #bundle1 B INNER JOIN Ref_L...	4.4%	4	12	9	
SELECT @IsTom9116Enabled = [Enabled] FROM dbo.FeatureToggle WHE...	0.0%	0	9	0	
UPDATE B SET FrontOfficeCode = RFOC.label FROM #bundle1 B INNER J...	4.3%	4	9	7	
INSERT INTO #bundle1 (idBundle, idBundleUp, lft, rgt, idTree, valueDat...	4.2%	14	8	8	
INSERT INTO #bundle1 (idBundle, idBundleUp, lft, rgt, idTree, valueDat...	4.1%	10	7	6	
UPDATE #bundle1 SET type = 'CombinedFacility', idItemType = 17, -- Re...	1.5%	14	4	4	
UPDATE #bundle1 SET ca3isodevEOD = ISNULL(CF.Currency, C.ca3isod...	1.3%	8	3	3	
INSERT INTO @CombinedFacility SELECT DISTINCT m.idBundle, m.idBBM...	0.8%	4	2	2	
INSERT INTO @foCodesTable SELECT * FROM fn_GetDistinctFOCodes(...	0.1%	0	1	1	
SELECT * INTO #bundle1 FROM dbo.ts_Bundle_create()	0.1%	1	1	1	
UPDATE #bundle1 SET ca3isodevEOD = ISNULL(ca3isodevEOD, ca3isode...	0.7%	3	1	1	
IF (@idBundle IS NOT NULL)	0.0%		0	0	
IF (@IsTom9116Enabled = 1)	0.0%		0	0	
IF (@isActive IS NULL)	0.0%		0	0	
SET @isActiveCondition = N' ((@isActiveParam = 1 AND (F.idFacilityStat...	0.0%		0	0	
IF (@localizationSG IS NULL)	0.0%		0	0	
SET @localizationSGCondition = N' ';	0.0%		0	0	
IF (@idFOCode IS NULL)	0.0%		0	0	
SET @idFOCodeCondition = N' ';	0.0%		0	0	
IF (@isMigrated IS NULL)	0.0%		0	0	

dbo.tmp_testBundleFilter_For_Val @valueDate = '2020-11-03 11:59:52', -- date...	100.0%		1,807	188	100.0%
EXEC val.[prv_BundleFilter_set] @valueDate = @valueDate, -- datetime @id...	99.1%		1,476	172	98.9%
INSERT INTO #bundleLocal (idBundle, idBundleUp, idItemType, [type], v...	30.7%	47	76	74	54.6%
UPDATE @CombinedFacility SET isActive = 1 WHERE idBBMultiFacilityHea...	0.6%	36	39	29	0.1%
INSERT INTO #bundleLocal (idBundle, idBundleUp, idItemType, [type], v...	21.7%	146	15	13	19.9%
INSERT INTO #bundleLocal (idBundle, idBundleUp, idItemType, [type], v...	13.8%	32	10	9	10.4%
INSERT INTO #bundleLocal (idBundle, idBundleUp, idItemType, [type], v...	21.9%	49	10	10	11.6%
INSERT INTO @CombinedFacility SELECT DISTINCT m.idBundle, m.idBBM...	1.0%	4	7	1	1.2%
INSERT INTO #bundle (idBundle, idBundleUp, idItemType, [type], value...	8.5%	0	3	2	0.1%
INSERT INTO @foCodesTable SELECT * FROM fn_GetDistinctFOCodes(...	0.1%	0	1	1	0.0%
DROP TABLE #authorizedFacilities;	0.0%		1	0	0.0%
DECLARE @CUSTOMER_ITEM_TYPE INT = 1;	0.0%		0	0	0.0%
DECLARE @FACILITY_ITEM_TYPE INT = 2;	0.0%		0	0	0.0%
DECLARE @BOX_ITEM_TYPE INT = 3;	0.0%		0	0	0.0%
DECLARE @COMBINED_FACILITY_ITEM_TYPE INT = 17;	0.0%		0	0	0.0%
DECLARE @TRANSACTION_STATUS_TEMPLATE INT = 4;	0.0%		0	0	0.0%
DECLARE @FACILITY_STATE_ACTIVE INT = 1;	0.0%		0	0	0.0%
DECLARE @FACILITY_STATE_INACTIVE INT = 0;	0.0%		0	0	0.0%
DECLARE @isActive INT = 1;	0.0%		0	0	0.0%
DECLARE @MaxDate DATETIME = '9999-01-01';	0.0%		0	0	0.0%
CREATE TABLE #bundleLocal (idBundle INT, idBundleUp INT, idItemType ...	0.0%		0	0	0.0%
CREATE TABLE #authorizedFacilities ([idFacility] INT NOT NULL, [idBundl...	0.0%		0	0	0.0%
IF (@idBundle IS NOT NULL)	0.0%		0	0	0.0%
IF OBJECT_ID('tempdb..#authorizedFacilities') IS NOT NULL	0.0%		0	0	0.0%
SELECT idBundle, idBundleUp, idItemType, valueDate, lft, rgt, idTree FROM ...	0.8%	0	211	2	1.1%
SELECT * INTO #bundle FROM val.ts_Bundle_create_temp_table()	0.1%	0	2	0	0.0%
RETURN	0.0%		0	0	0.0%
DECLARE @isActive INT = 1;	0.0%		0	0	0.0%

1

WHILE MAKING
CHANGES
TO EXISTING CODE

2

WHILE ADDING
NEW CODE

TDD
IS MUST



WHEN YOUR CODE WORKS THE FIRST TIME



IT'S JUST **HALF** THE JOB DONE

THE TIME THAT THE CODE FIRST
WORKS...

**...IT'S THE MOMENT IN WHICH
YOU START BUILDING YOUR
CAREER**

**...IT'S THE MOMENT IN WHICH
YOU START BUILDING YOUR
DESIGN SKILLS**

DOUBTS



REMORSE



**MICRO
DECISIONS**



**SELF
REVIEW**



IT'S PERFECT MOMENT TO
CLEAN THE CODE

NAMES	FUNCTIONS	COMMENTS
FORMATTING	OBJECTS & DATA STRUCTURES	ERROR HANDLING
CONTINUOUS REFACTORING	PAIR PROGRAMMING	MONITORING

CLEAN CODE TOPICS

NAMES

THE **POWER** INVESTED IN OUR
FINGERTIPS





A word cloud of Python-related terms. The words are arranged in a roughly circular pattern, with 'class' being the largest and most central. Other prominent words include 'global variable', 'instance variable', 'local variable', 'private function', 'public function', 'argument', 'namespace', 'module', 'library', 'package', 'program', 'file', and 'directory'. The words are in various shades of gray and sizes, creating a visual hierarchy.

local variable private function

instance variable

argument directory

class

global variable public function

namespace

module

library

package

program

file

...BECAUSE WE NAME **SO**
MUCH

...WE MUST USE THIS
POWER WISELY
WE MUST **DO IT WELL**



TIM OTTINGER

COINED THE PHRASE,
“**IMPLICITY** OF A CODE”

USE INTENTION-REVEALING NAMES

```
int d; // elapsed time in days
```



```
int elapsedTimeInDays;
```

```
public List<int[]> getThem() {  
    List<int[]> list1 = new ArrayList<int[]>();  
    for (int[] x : theList)  
        if (x[0] == 4)  
            list1.add(x);  
    return list1;  
}
```



```
public List<int[]> getFlaggedCells() {  
    List<int[]> flaggedCells = new ArrayList<int[]>();  
    for (int[] cell : gameBoard)  
        if (cell[STATUS_VALUE] == FLAGGED)  
            flaggedCells.add(cell);  
    return flaggedCells;  
}
```

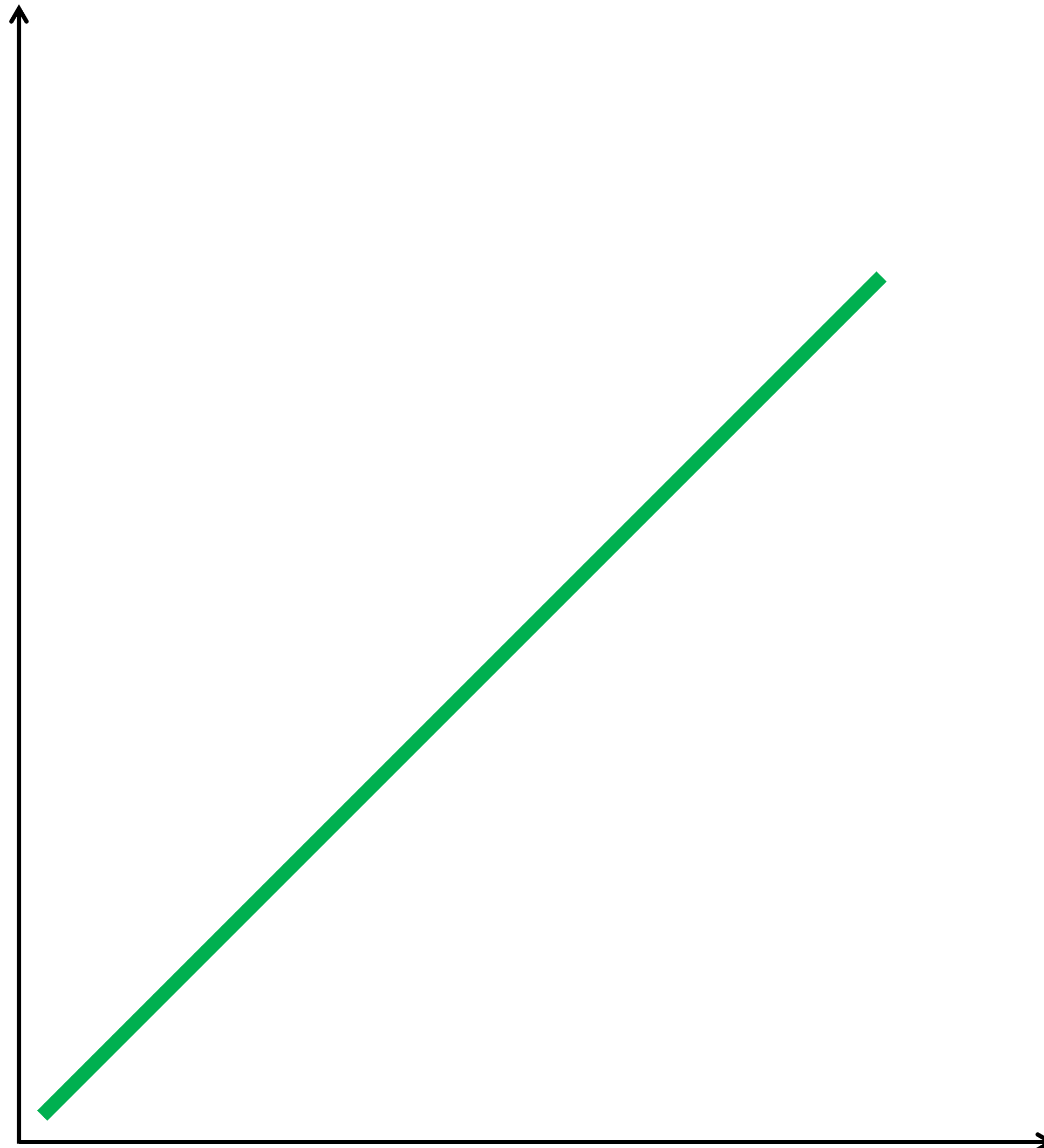
```
public List<Cell> getFlaggedCells() {  
    List<Cell> flaggedCells = new ArrayList<Cell>();  
    for (Cell cell : gameBoard)  
        if (cell.isFlagged())  
            flaggedCells.add(cell);  
    return flaggedCells;  
}
```

WE **COMMUNICATE**

LENGTH

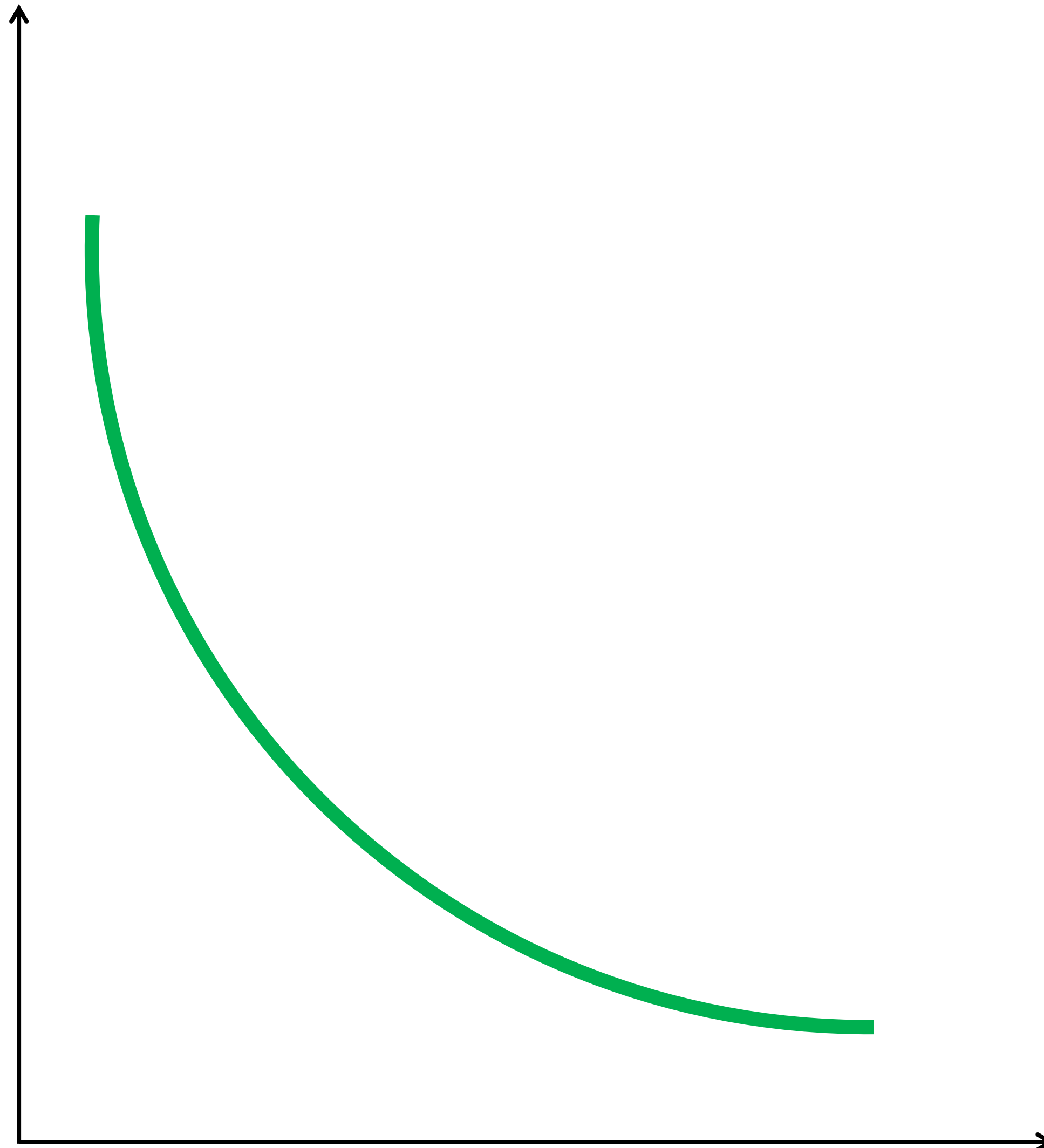


VARIABLE NAME
LENGTH



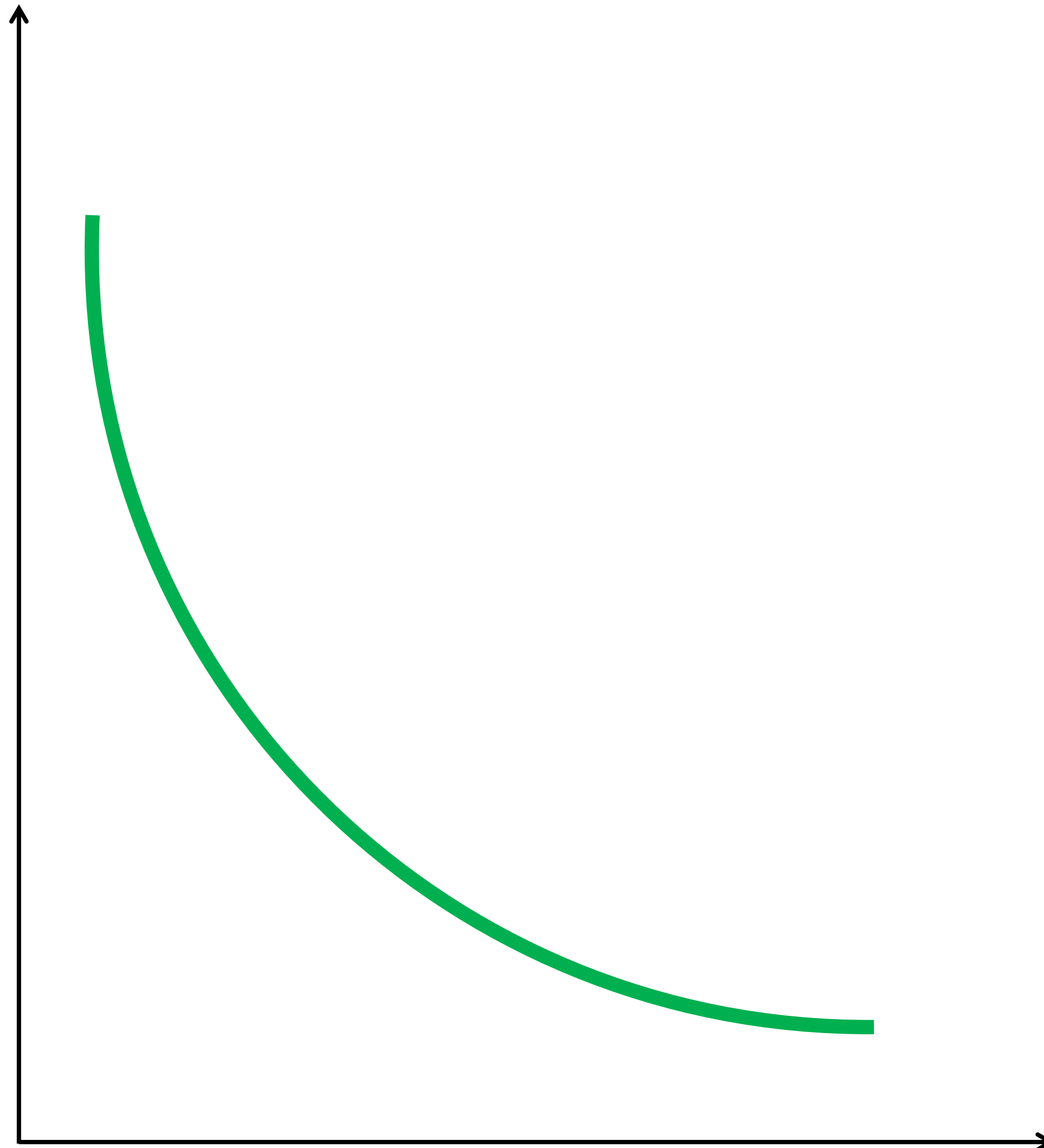
IT'S **SCOPE**

FUNCTION NAME
LENGTH










IT'S **SCOPE**

CLASS NAME
LENGTH



















IT'S **SCOPE**






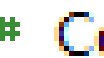















V 1.0

- ▶  Tomcat.StoredProcedure
- ▶  Properties
- ▶  References
- ▶  App.config
- ▶  Form1.cs
- ▶  Program.cs
- ▶  Utility.cs

V 2.0

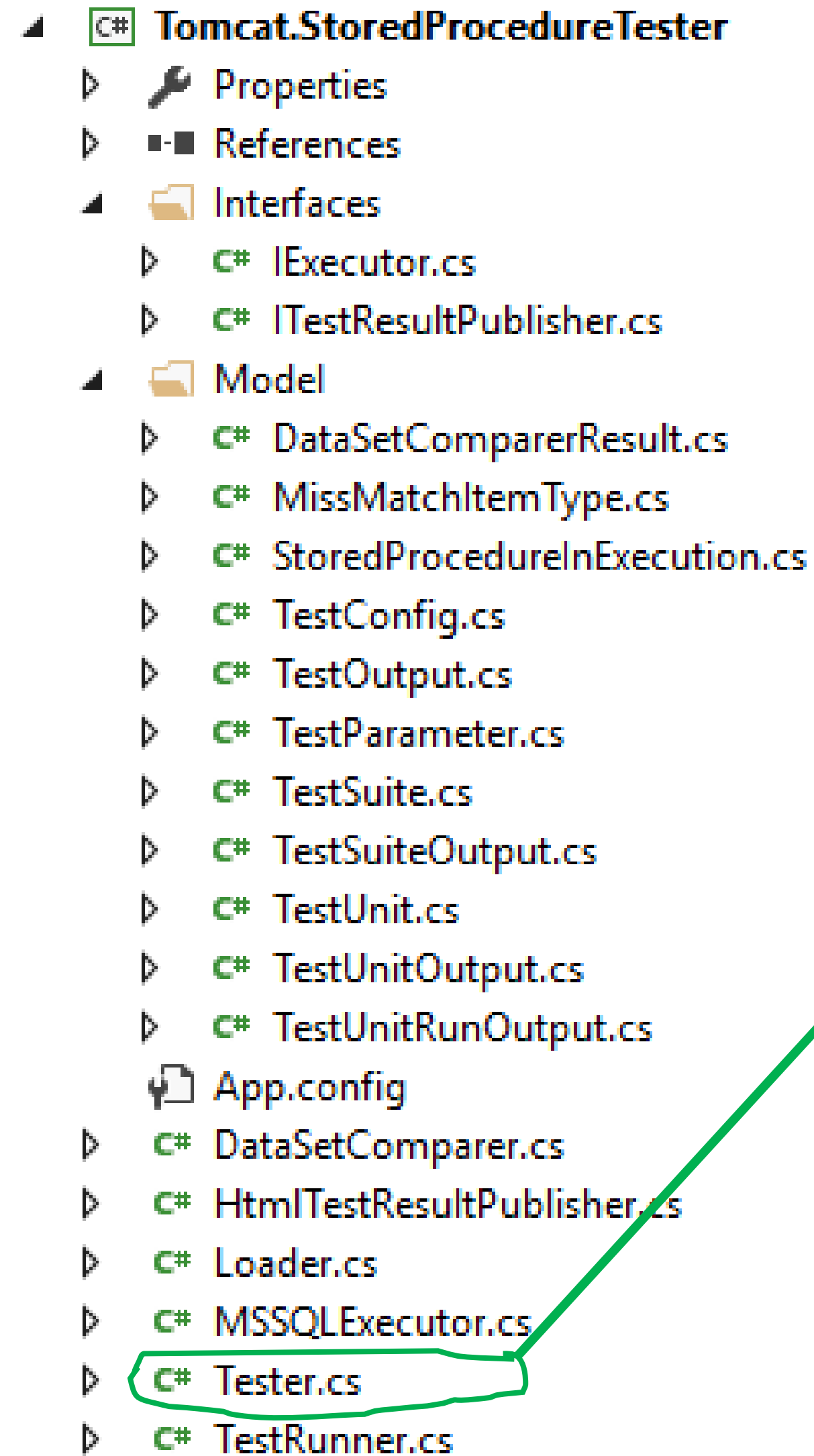
- ▶  Tomcat.SPComparer
- ▶  Properties
- ▶  References
- ▶  Config
 - ▶  SPs
 - ▶  serverconfig.json
 - ▶  App.config
- ▶  DataSetComparer.cs
- ▶  DBItem.cs
- ▶  Logger.cs
- ▶  packages.config
- ▶  Program.cs
- ▶  SPComparer.cs
- ▶  SPComparerOutput.cs
- ▶  StoredProcedureComparer.cs
- ▶  Utility.cs

V 3.0

- ▶  Tomcat.SPComparer
- ▶  Tomcat.SPComparerProvider
 - ▶  Properties
 - ▶  References
 - ▶  CommandOutput.cs
 - ▶  ComparerConfiguration.cs
 - ▶  ComparerProvider.cs
 - ▶  ComparingItem.cs
 - ▶  ComparingItemCollection.cs
 - ▶  DataSetComparer.cs
 - ▶  DataSetComparerResult.cs
 - ▶  DBItem.cs
 - ▶  ExecutionOutput.cs
 - ▶  ExecutionTime.cs
 - ▶  IStoredProcedureExecutor.cs
 - ▶  Logger.cs
 - ▶  SPComparerOutput.cs
 - ▶  StoredProcedureComparer.cs
 - ▶  StoredProcedureExecutor.cs
 - ▶  StoredProcedureOutputComparer.cs
 - ▶  Utility.cs

V 4.0

???



THE MOMENT NAME
WAS **GIVEN** TO THIS
CLASS THE WHOLE
THING BECAME
CRYSTAL CLEAR

GOOD NAMES TELLS
THE **CONTEXT** OF
ENTIRE SYSTEM

WE COMMUNICATE

AVOID DISINFORMATION

```
10 var XYZFooBarClassForBlabla
11 var XYZFooBarClassForBlablable
12
```

```
4 var a = 1;
5 if ( 0 == 1 )
6 a = 01;
7 else
8 1 = 01;
9
```

```
public class Document
public class DocumentInfo
public class DocumentDetail
public class Collateral

public class Documents
public class DocumentList
```

Using lower case **l**
(looks like **number-1**) and
uppercase **O** (looks like
number-**0**) are also
unhelpful.

“A software author must avoid leaving **false clues**
which obscure the meaning of code.”
- Ottinger

WE **COMMUNICATE**

USE PRONOUNCEABLE NAMES

```
// generation date, year, months, day, hour, minute, and second  
class DtaRcrd102 {  
    private Date genymdhms;  
    /* ... */  
}
```



```
// better:  
class Customer {  
    private Date generationTimestamp;  
    /* ... */  
}
```

```
@amount float,  
@ca3isodev char(3),  
@valueDate datetime,  
@expiryDate datetime ,
```



CURRENCY

```
public void CalculateSCPUAvailableQuantity(DocumentTransfer transfer)
```

“Programming is a social activity” - Bob Martin

WE **COMMUNICATE**

USE SEARCHABLE NAMES

```
for (int j=0; j<34; j++) {  
    s += (t[j]*4)/5;  
}
```



```
int realDaysPerIdealDay = 4;  
const int WORK_DAYS_PER_WEEK = 5;  
int sum = 0;  
for (int j=0; j < NUMBER_OF_TASKS; j++) {  
    int realTaskDays = taskEstimate[j] * realDaysPerIdealDay;  
    int realTaskWeeks = (realdays / WORK_DAYS_PER_WEEK);  
    sum += realTaskWeeks;  
}
```

```
let a = {'apple' : 2, 'mango' : 1,  
        'banana': 0, 'orange': 2,  
        'water-melon': 2}  
  
for (b in a) {  
    let c = 'no'  
    if (a[b] == 2) c = 'many'  
    else if (a[b] == 1) c = 'one'  
    else if (a[b] == 0) c = 'no'  
  
    console.log(a, 'has', c, c == 'many'? 'seeds' : 'seed')  
}
```



```
const NO_SEED = 0  
const ONE_SEED = 1;  
const MANY_SEEDS = 2;  
  
let fruits = {'apple' : MANY_SEEDS, 'mango' : ONE_SEED,  
             'banana': NO_SEED, 'orange': MANY_SEEDS,  
             'water-melon': MANY_SEEDS}  
  
// print the seeds count  
for (fruit in fruits) {  
    let count = 'no';  
  
    if (fruits[fruit] == MANY_SEEDS) count = 'many'  
    else if (fruits[fruit] == ONE_SEED) count = 'one'  
  
    console.log(fruit, 'has', count, count == 'many'? 'seeds' : 'seed')  
}
```

Replace literals with constants

CLASS NAMES

A CLASS NAME SHOULD
BE A **NOUN**,
NOT A **VERB**.

AVOID WORDS like
Manager, Processor, Data, or Info.

GOOD NAMES could be:
Tester, TestSuite, TestUnit, HtmlTestResultPublisher.

METHOD NAMES

METHODS SHOULD HAVE **VERB**

```
internal static void Consolidation(Process process,  
internal static void UnitDdaNetting(  
internal static void UnitAnalyticalOutstandingGeneration(Process process,  
public double QualityUnitConversion(int quality, int idUnit, double quantity,  
                                     int sourceDocumentIdQuality,  
                                     int sourceDocumentIdUnit,  
                                     DateTime valueDate)...
```

GOOD NAMES could be:

```
Save(), Run(), Publish(), Valuate(),  
Consolidate()  
RunNettingValuation(),  
GenerateAnalyticalOutstanding()  
ConvertQualityUnit()
```

WE **COMMUNICATE**

HOWEVER

THERE ARE **NO** GOOD NAMES

IT'S CONTINEIOUS PROCESS
EVERY TIME YOU HAVE BETTER
UNDERSTANDING OF CODE,
YOU FIND A BETTER NAME

ENGLAND'S ETON COLLEGE



“THAT’S EASY”



JUST BRUSH
OFF THE DEW
EVERY MORNING



MOW THEM
**EVERY
OTHER DAY**



AND ROLL
THEM
ONCE A WEEK



DO THAT FOR **500 YEARS** &
YOU'LL HAVE A NICE LAWN, TOO



CARE

tl;**dr**

NEGLECTION **ACCELERATES** ROT
FASTER THAN ANY OTHER FACTOR

TEST AUTOMATION GIVES YOU
POWER OF
FEARLESS REFACTORING

FIND **OPPORTUNITIES** OF
AUTOMATION IN ABSENCE OF
TESTS FOR OLD CODE

USE THE **GIFTED** POWER OF
NAMING THINGS WISELY

USE **INTENTION-REVEALING** NAMES

AVOID **DISINFORMATION**

USE **PRONOUNCEABLE** NAMES

USE **SEARCHABLE** NAMES

CLASS NAMES MUST BE **NOUNS**
METHOD NAMES MUST BE **VERBS**

REMEMBER WE **DON'T CODE**,
WE **COMMUNICATE**

REMEMBER WE **DON'T CODE**,
WE **COMMUNICATE**

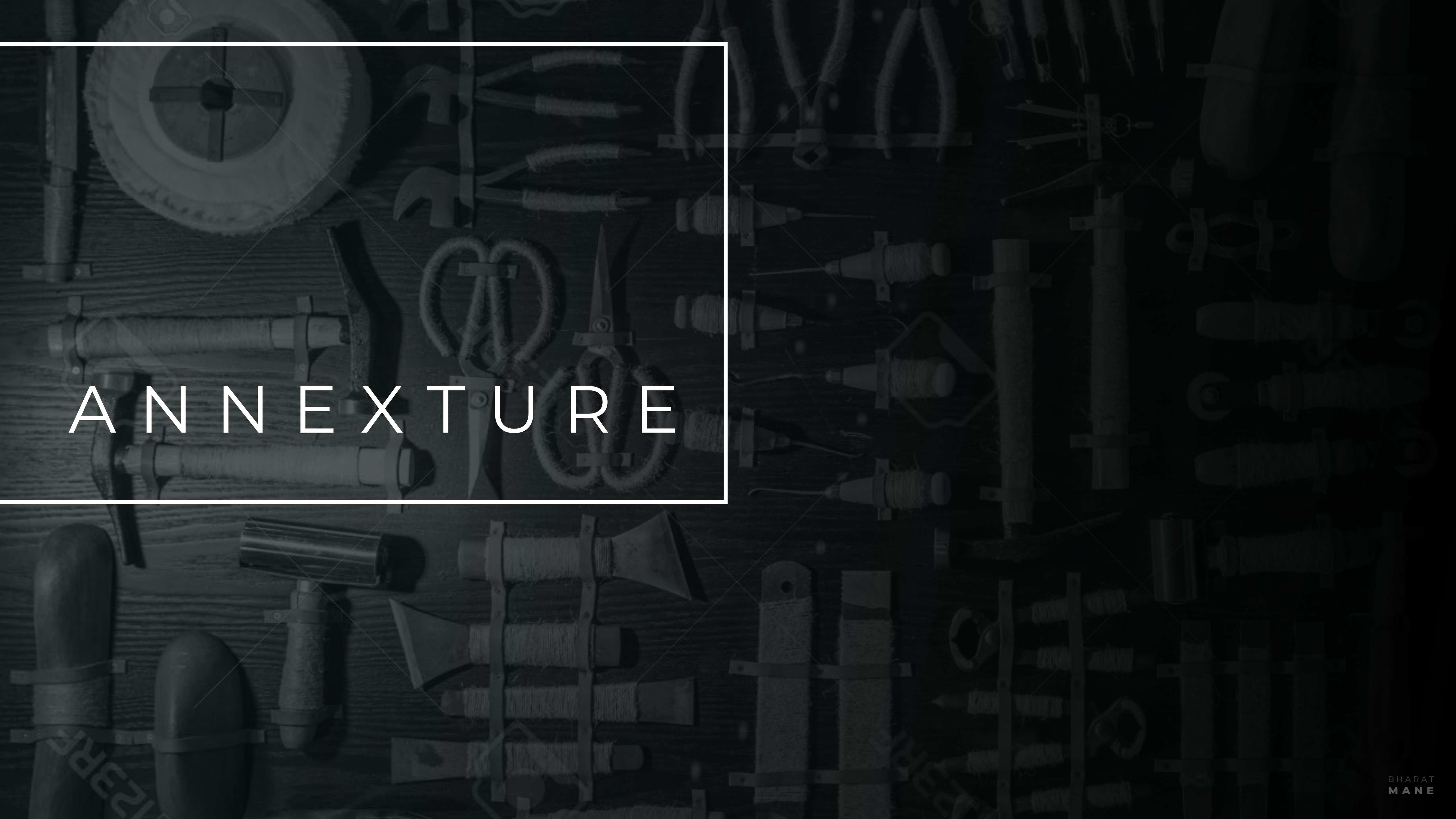
DON'T FEAR
&
TAKE CARE

MINDSET

WANT TO DO IT

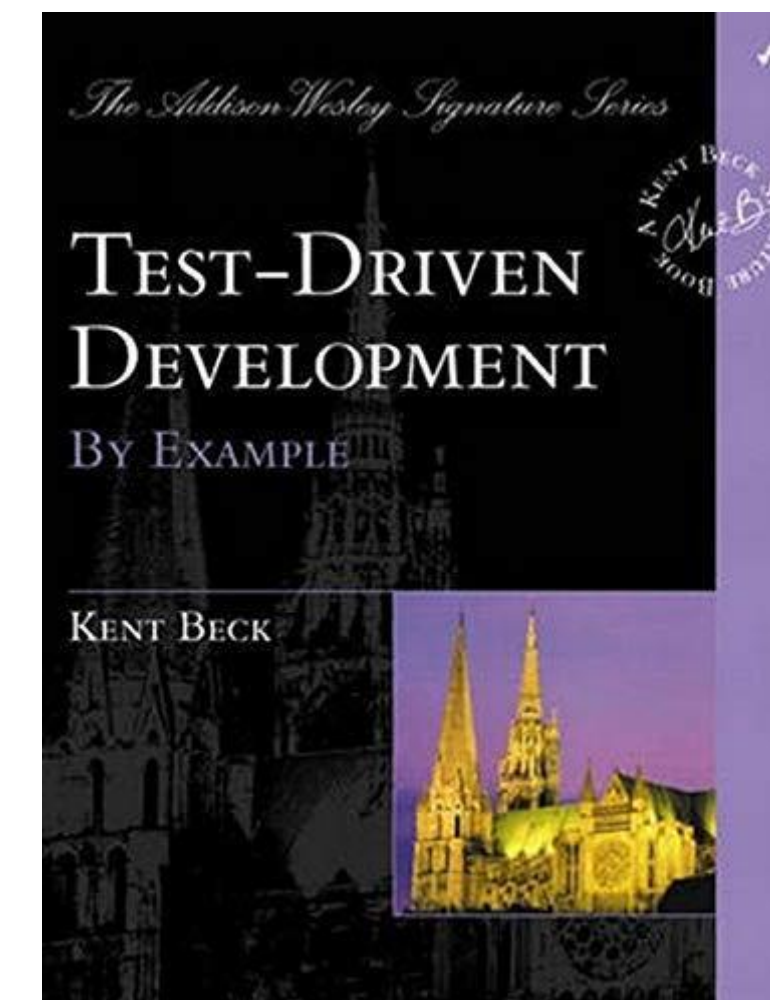
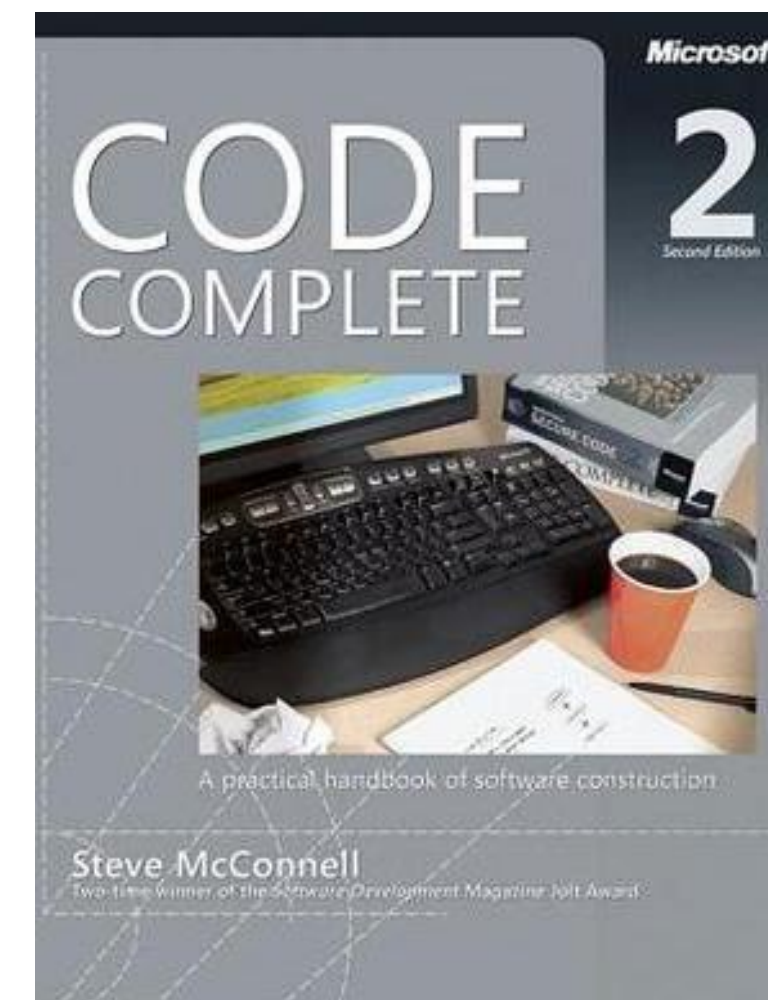
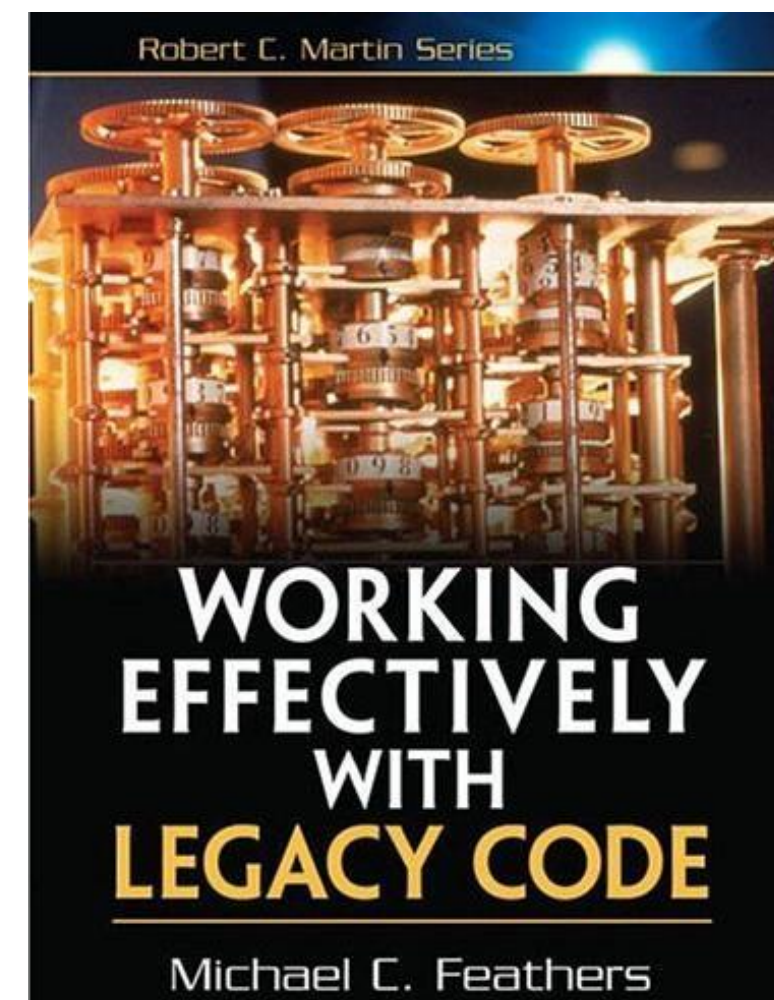
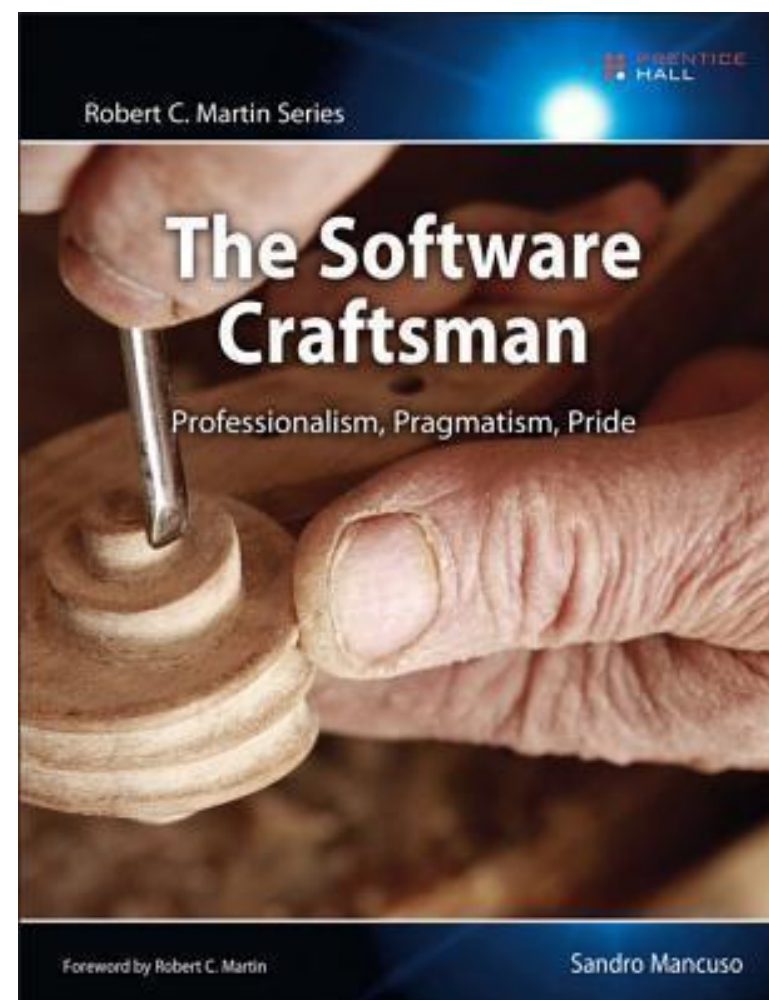
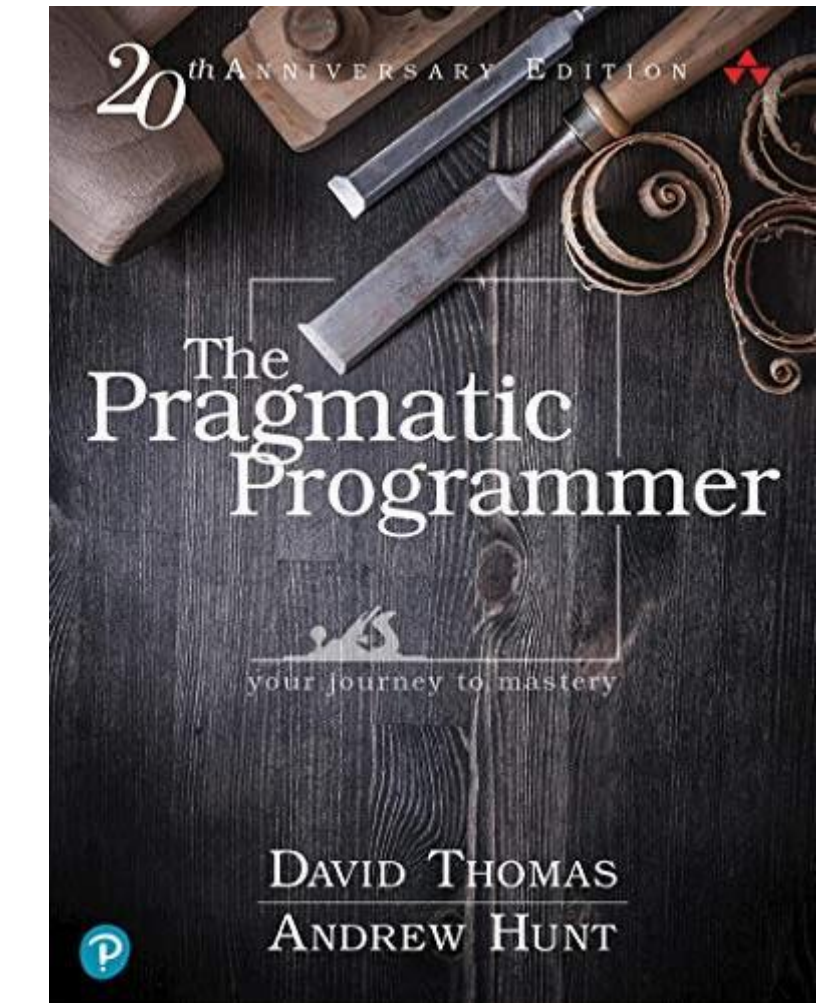
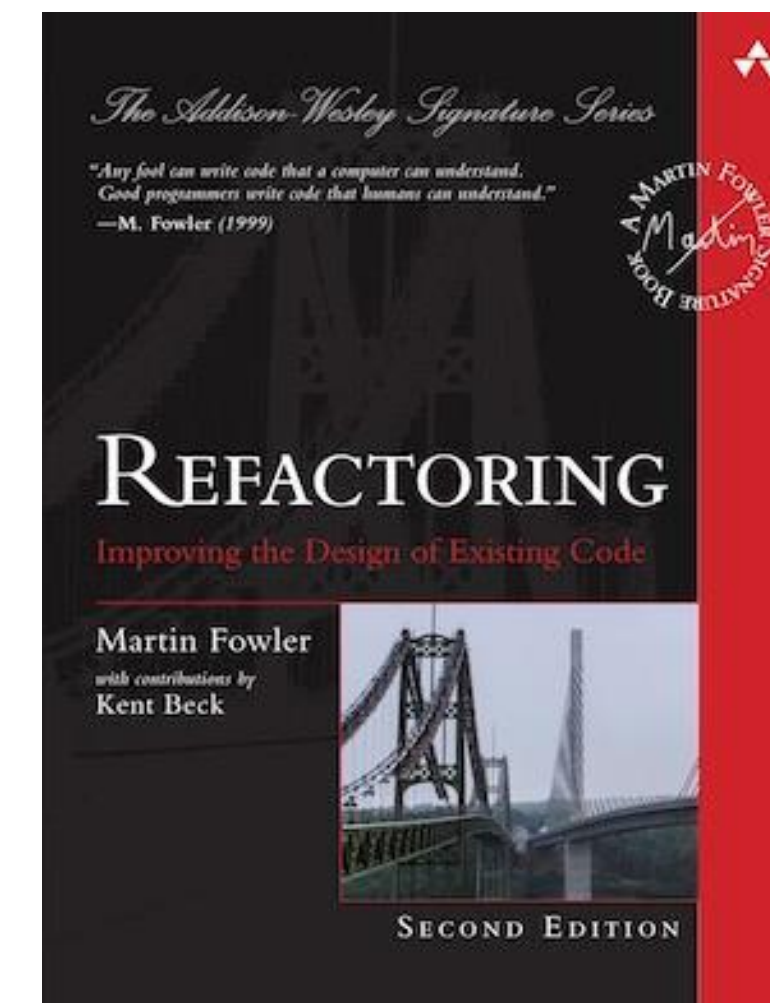
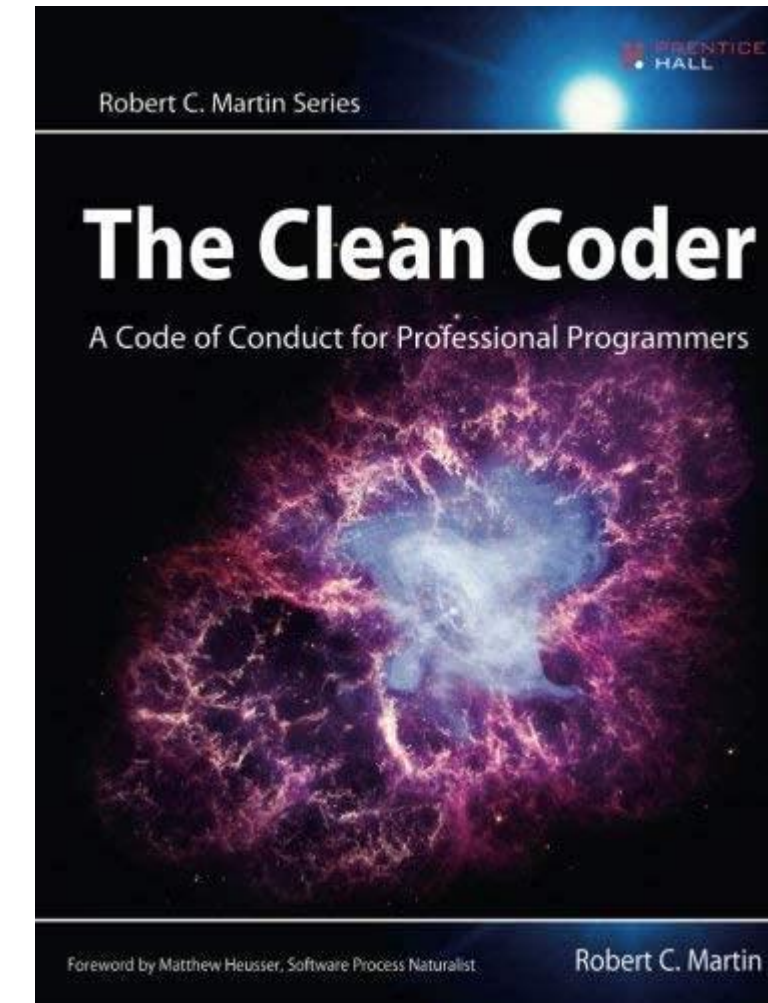
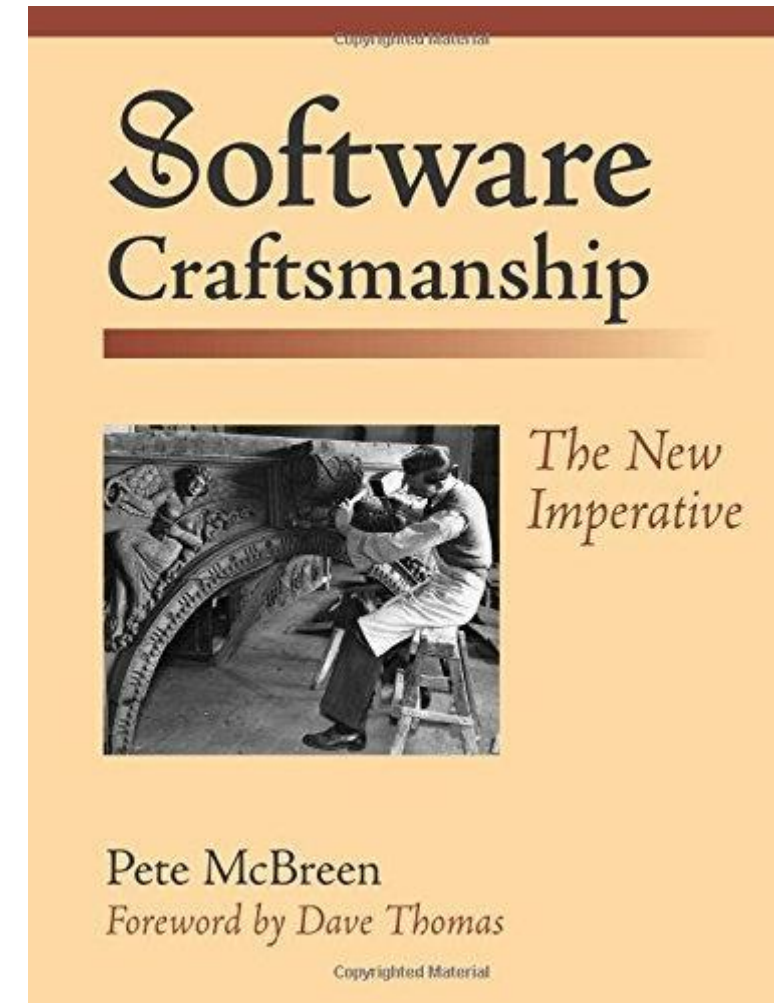
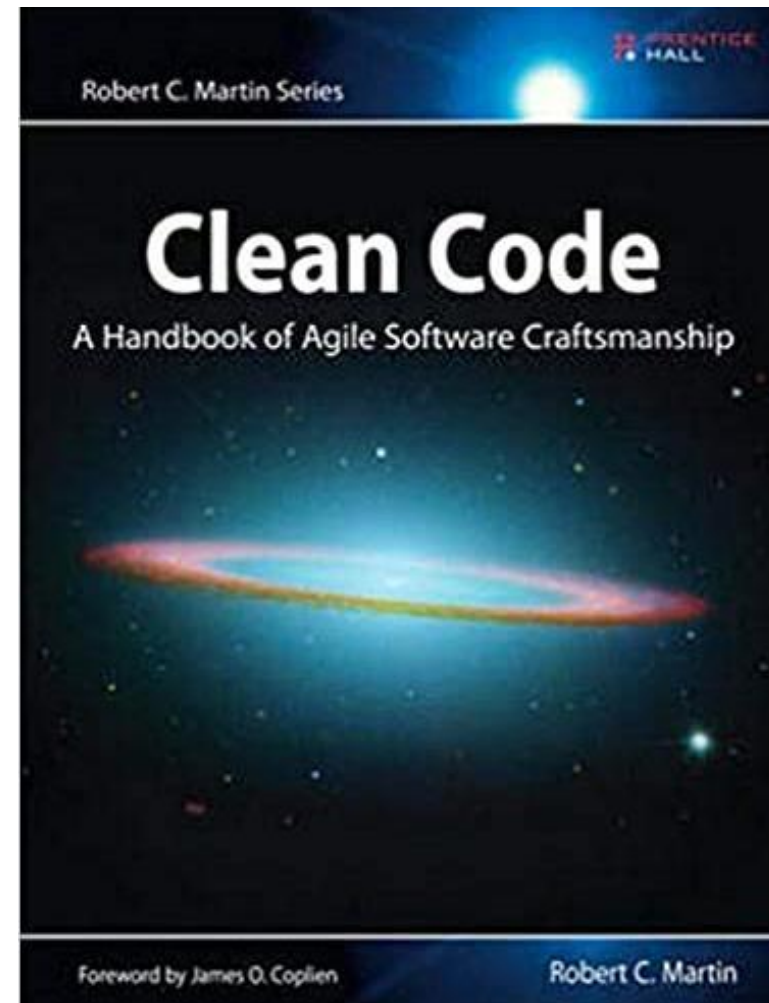


Q & A

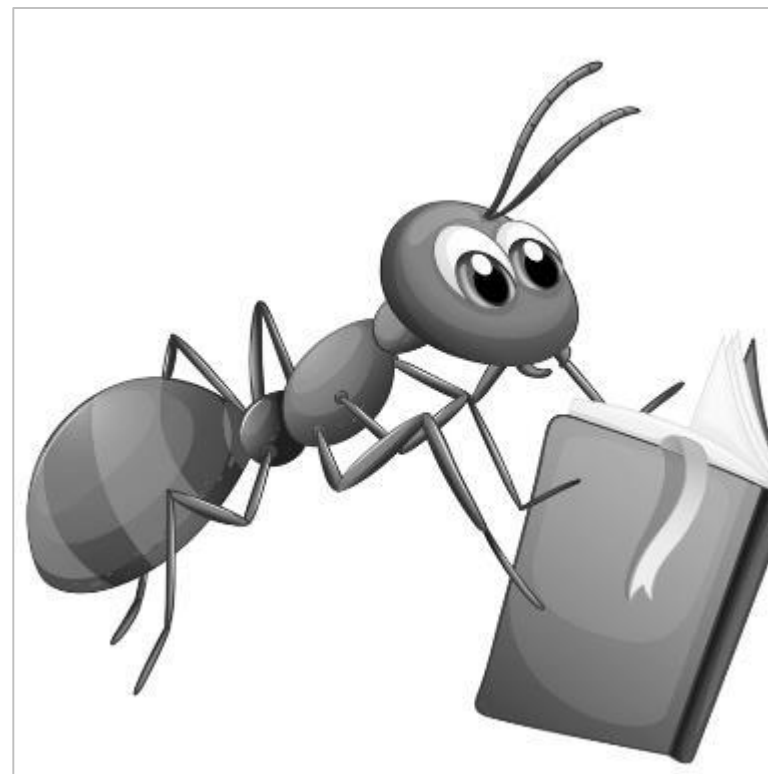


ANNEXTURE

GOOD READS



principles
patterns
practices
heuristics



KNOWLEDGE

+

Grind that
knowledge
into muscle
memory



WORK

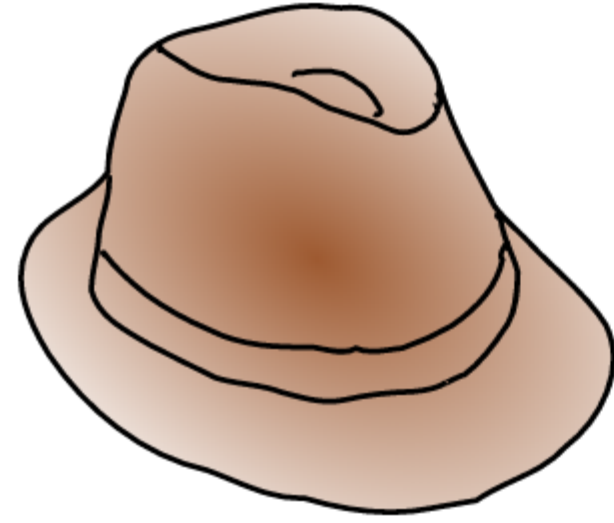
=



CRAFTSMANSHIP

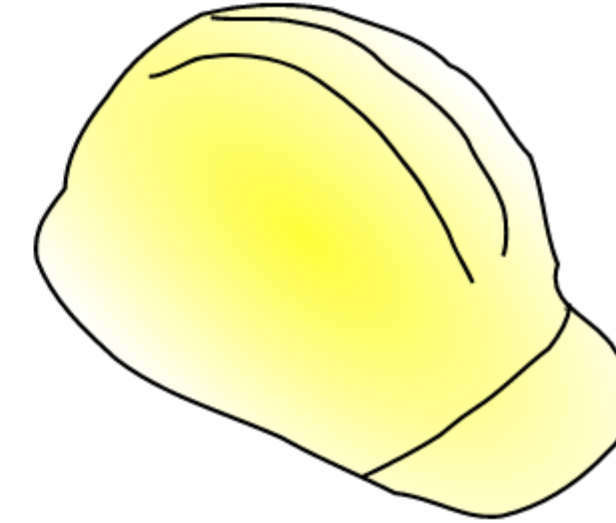
“HONESTY IN SMALL THINGS
IS NOT A SMALL THING.”

-Danish Saw



Refactoring

When refactoring every change you make is a small behavior-preserving change. You only refactor with green tests, and any test failing indicates a mistake. By stringing together a series of small changes like this you can move more quickly and with less risk because you shouldn't get trapped in debugging.



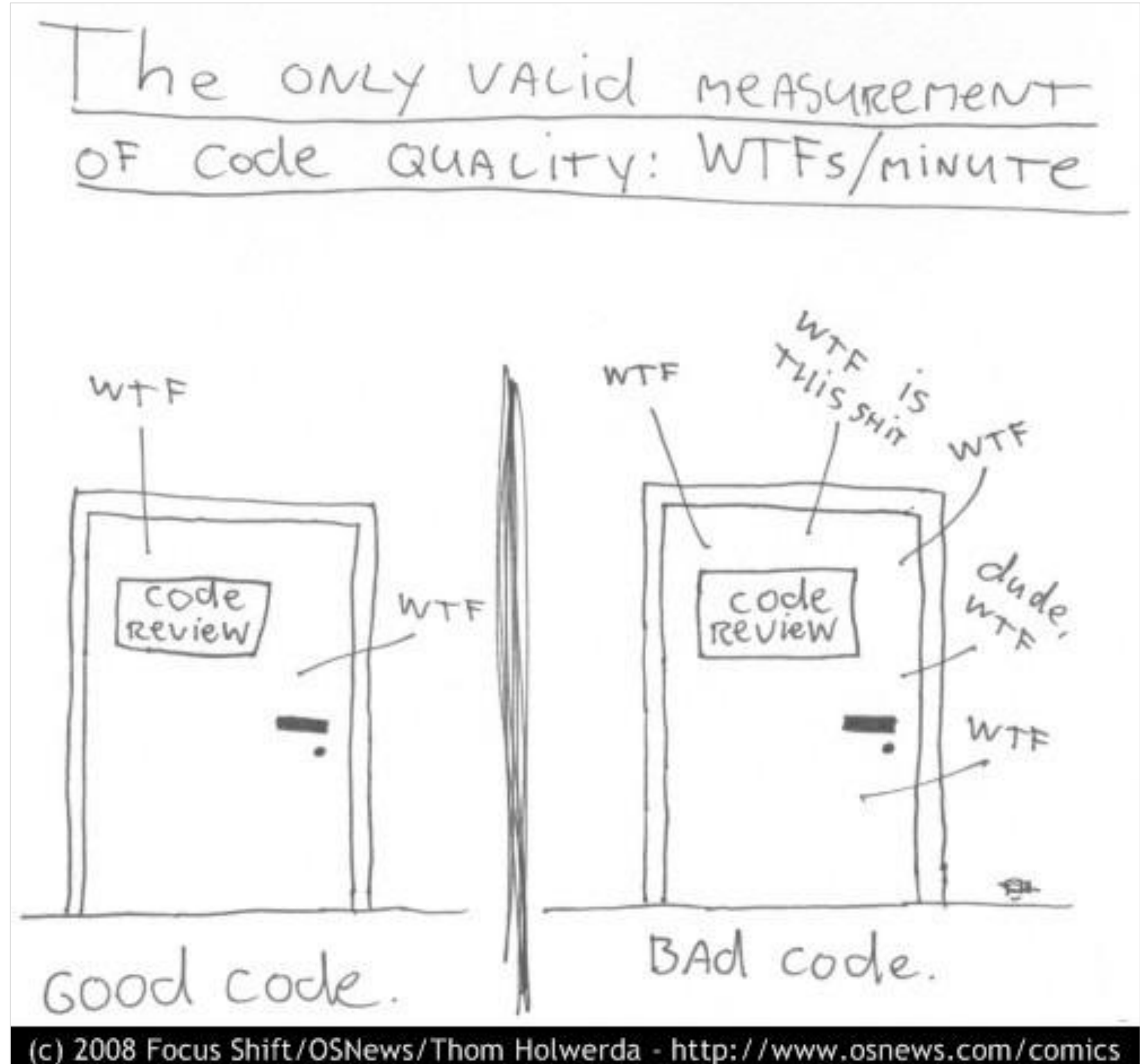
Adding Function

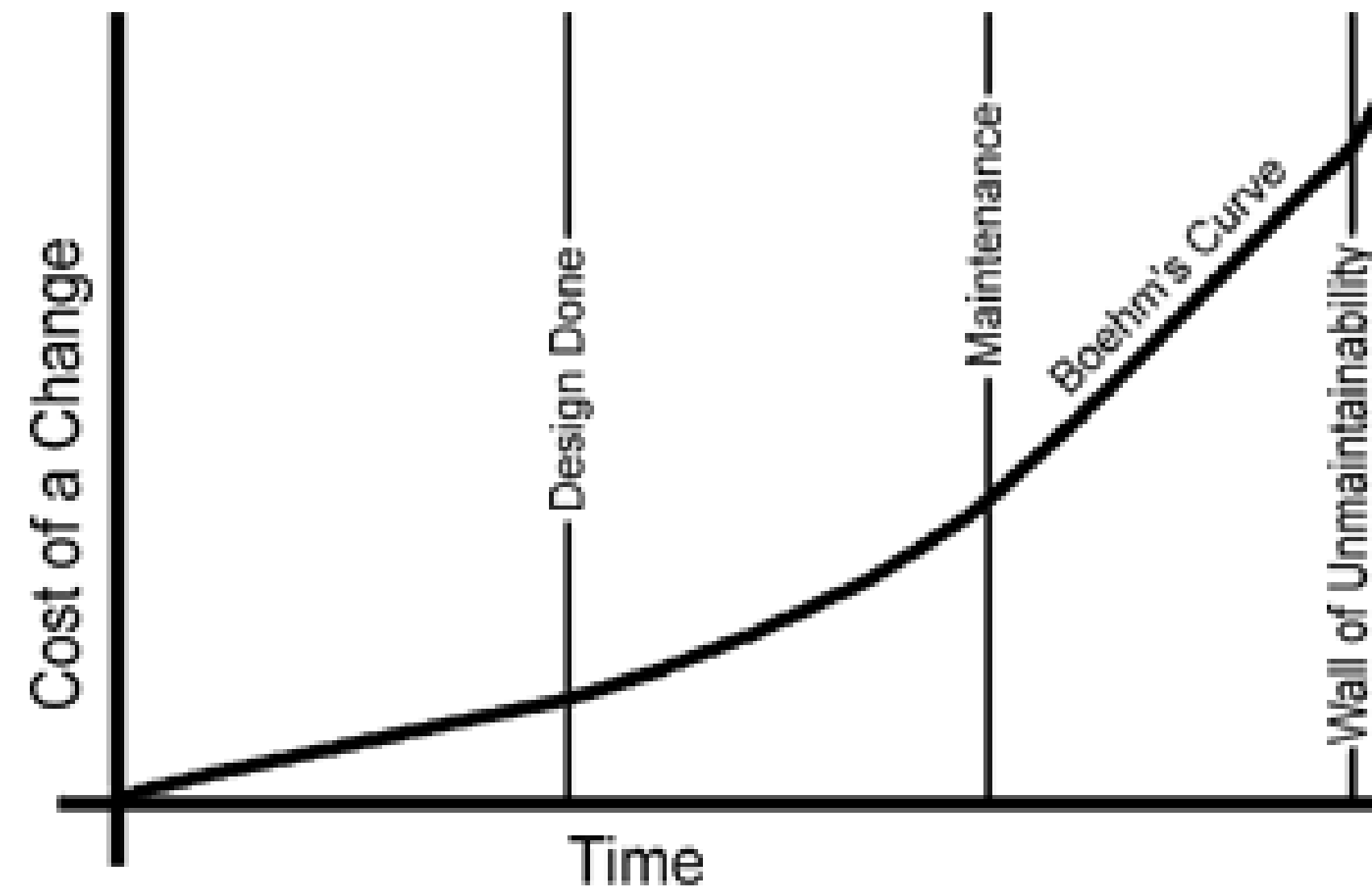
Any other change to the code is adding function. You will add new tests and break existing tests. You aren't confined to behavior-preserving changes (but it's wise to keep changes small and return to green tests swiftly).

During programming you may swap frequently between hats, perhaps every couple of minutes. But...

You can only wear one hat at a time

UNIT OF CODE QUALITY



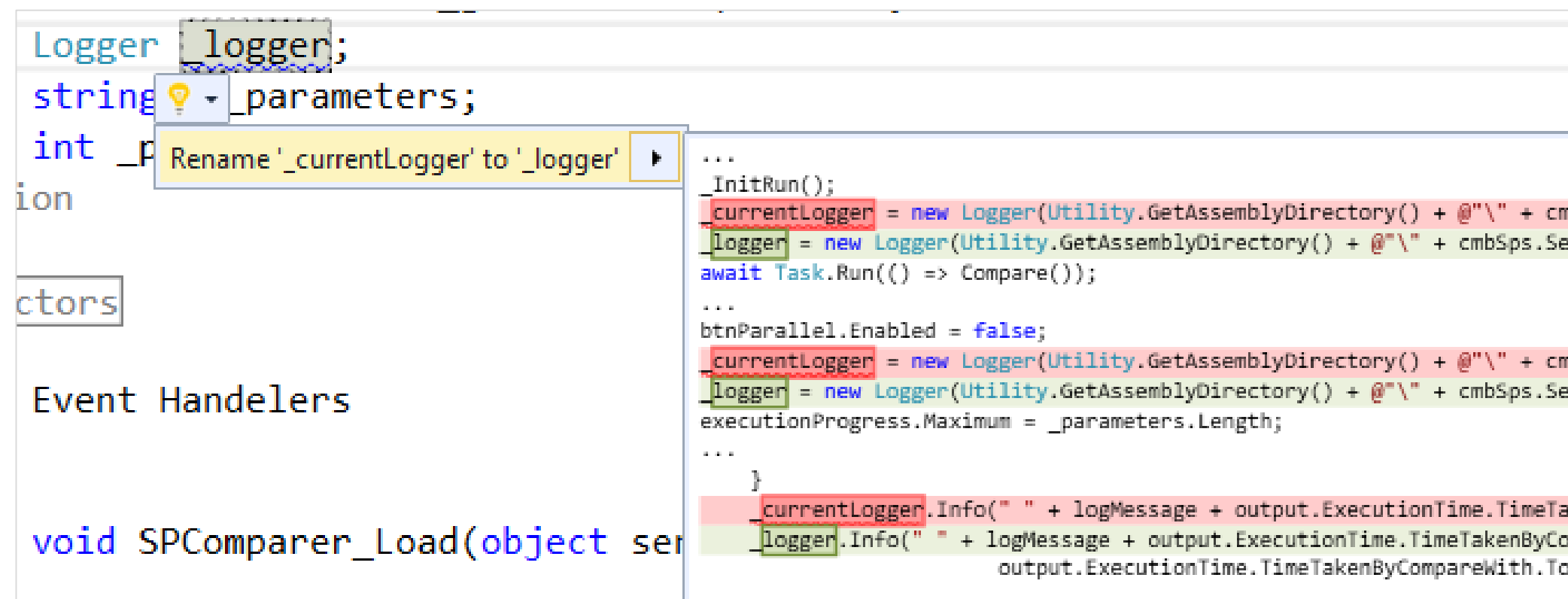


When we proclaim the design is done and accept no more changes

When we move the system into maintenance and change the team's process

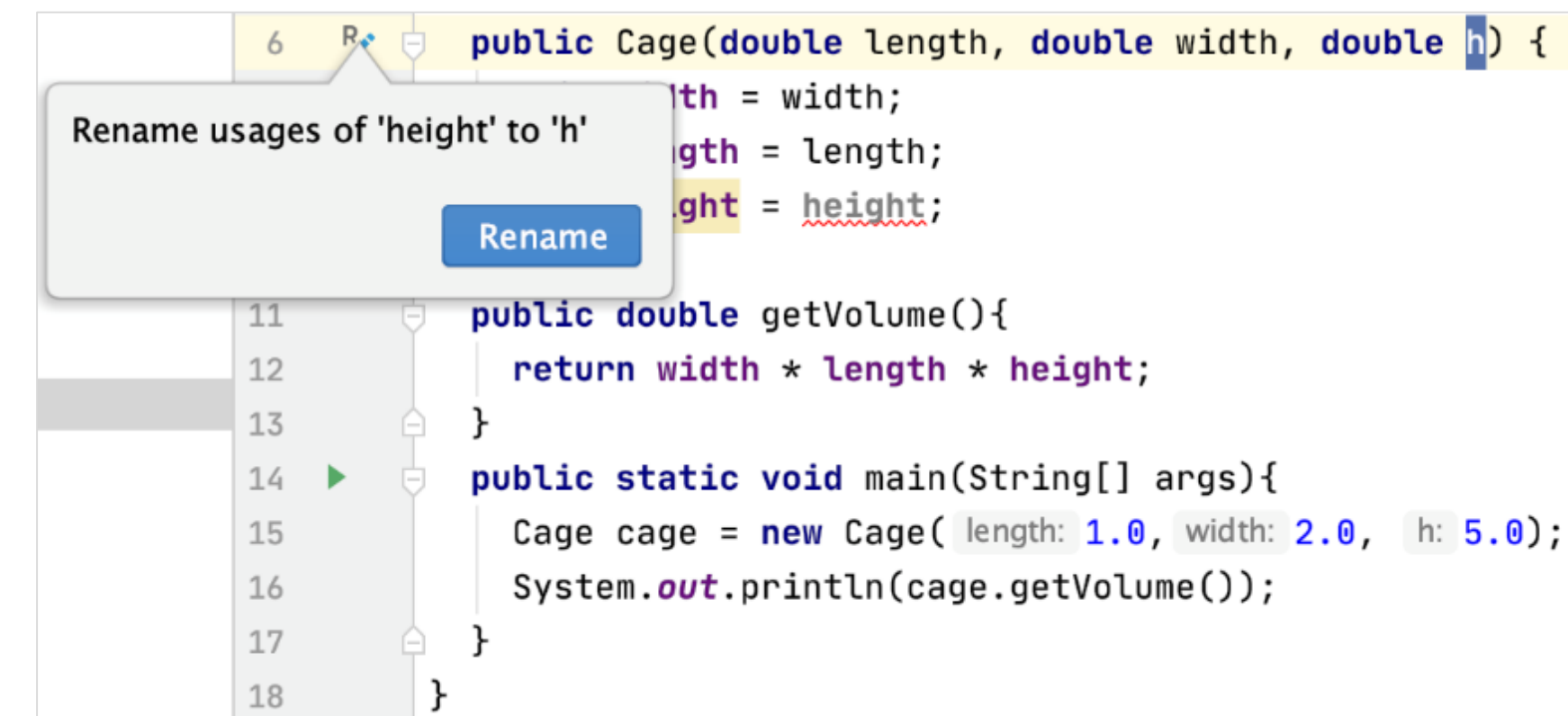
THERE ARE ONLY TWO HARD THINGS IN
COMPUTER SCIENCE:
CACHE INVALIDATION AND NAMING THINGS

MOST OF THE IDES GIVES YOU VERY
SAFE WAY TO RENAME
VARIABLES, CLASSES, FUNCTIONS
ESPECIALLY PRIVATE STUFF IS
COMPLETELY SAGE



This screenshot shows a Visual Studio code editor with a C# file. A variable named `_logger` is being renamed from `_currentLogger`. A context menu is open over the variable, showing the option "Rename '_currentLogger' to '_logger'". The code includes a `Logger` class and a `SPComparer_Load` method. The `Logger` class has a `Log` method. The `SPComparer_Load` method calls `Log` on a `currentLogger` object. The `Log` method is highlighted in red, and the `currentLogger` variable is highlighted in green. The `Log` method is highlighted in red, and the `currentLogger` variable is highlighted in green.

```
Logger logger;  
string _parameters;  
int _P  
ion  
ctors  
Event Handelers  
void SPComparer_Load(object ser
```



This screenshot shows a Visual Studio code editor with a C# file. A variable named `height` is being renamed to `h`. A context menu is open over the variable, showing the option "Rename usages of 'height' to 'h'". The code includes a `Cage` class and a `main` method. The `Cage` class has a `getVolume` method. The `main` method calls `getVolume` on a `Cage` object. The `getVolume` method is highlighted in red, and the `height` variable is highlighted in green. The `getVolume` method is highlighted in red, and the `height` variable is highlighted in green.

```
6 public Cage(double length, double width, double h) {  
    th = width;  
    gth = length;  
    ght = height;  
11 public double getVolume(){  
12     return width * length * height;  
13 }  
14 public static void main(String[] args){  
15     Cage cage = new Cage( length: 1.0, width: 2.0, h: 5.0);  
16     System.out.println(cage.getVolume());  
17 }  
18 }
```

COMPREHENSION REFACTORING

Manifesto for Software Craftsmanship

Raising the bar.

As aspiring Software Craftsmen we are raising the bar of professional software development by practicing it and helping others learn the craft. Through this work we have come to value:

Not only working software,
but also **well-crafted software**

Not only responding to change,
but also **steadily adding value**

Not only individuals and interactions,
but also **a community of professionals**

Not only customer collaboration,
but also **productive partnerships**

That is, in pursuit of the items on the left we have found the items on the right to be indispensable.

© 2009, the undersigned.
this statement may be freely copied in any form,
but only in its entirety.

MANIFESTO FOR SOFTWARE CRAFTSMANSHIP

EXTREME PROGRAMMING PRACTICES

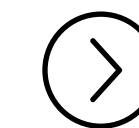
Group	Practices
Feedback	<ul style="list-style-type: none">✓ Test-Driven Development✓ The Planning Game✓ On-site Customer✓ Pair Programming
Continual Process	<ul style="list-style-type: none">✓ Continuous Integration✓ Code Refactoring✓ Small Releases
Code understanding	<ul style="list-style-type: none">✓ Simple Design✓ Collective Code Ownership✓ System Metaphor✓ Coding Standards
Work conditions	<ul style="list-style-type: none">✓ 40-Hour Week

XP has simple rules that are based on **5 values**.



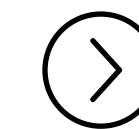
COMMUNICATION

Everyone on a team works jointly at every stage of the project



SIMPLICITY

Developers strive to write simple code bringing more value to a product, as it saves time and efforts.



FEEDBACK

Team members deliver software frequently, get feedback about it, and improve a product according to the new requirements.



RESPECT

Every person assigned to a project contributes to a common goal.



COURAGE

Programmers objectively evaluate their own results without making excuses and are always ready to respond to changes.